

- Combat-oriented MUDs (LP/Diku/etc, originally)
- Social-oriented MUDs (TinyMUD & its descendants, etc)
- Miscellaneous (mixture of the above, or hard to classify)

} types of MUDs

The majority of the muds in the miscellaneous category are not combat-oriented muds at all, and indeed many take after TinyMUD in most things. However, as these muds are not a direct derivative of the original TinyMUD code, I've stuck them in their own category. The authors listed for each server are very probably not the people currently working on that code. To find out who's currently in charge of the code, either ftp the latest version and look for a README file, or ask around.

- Combat

A note on the term combat-oriented: this generally means that combat is an inherent part of the culture of the mud. A flight-simulator could be called a combat-oriented game, just as truly as your typical shoot-em-up game could be. A social-oriented mud has a different focus, one dependent either on roleplaying social interactions (which MAY include combat!), or on not roleplaying at all, but merely talking with friends or other such benign things. It should be emphasized that simply because a given server is listed in the combat-oriented area, it does not necessarily follow that it *must* be a combat-oriented MUD. Most servers are fairly flexible, and can be used for social and combat uses alike, as well as for business and education. These categories are getting rather dated, and may be changed at some point in the future for ones that make more sense.

- Social

- business

- education

Detailed listings of the following servers are below. Note that the servers are organized **roughly** by type, and not by operating system. Most are designed for Unix, but several have been ported to other platforms, and will be noted as such in that server's entry. Directions for how to ftp and unarchive servers can be found at the end of this FAQ.

Combat-Oriented MUDs

AberMUD, LPMUD, DGD, DikuMUD, YAMA, UriMUD, Ogham, CircleMUD, AmigaMUD, Realms, Ursha Null 7

Social-Oriented MUDs

TinyMUD, TinyMUCK v1.* , TinyMUSH, PennMUSH, AlloyMUSH, TinyMUCK v2.* , TinyMUSE, TinyMAGE, MUG, TeenyMUD, TinyMUX

Misc MUDs

MUD, UberMUD, MOO, LambdaMOO, SMUG, UnterMUD, Mordor, COOLMUD, Cold Server

Combat-Oriented MUDs

AberMUD

One of the first adventure-based MUDs. Players cannot build. In later versions, a class system was added, and wizards can build onto the database. It's named after the university at which it was written, Aberystwyth. Latest version is 5.21.5. Supports all the usual in combat game design, including BSX graphics and MudWHO. Not too big, and it will run under BSD and SYSV. Amiga TCP/IP support now included.

Author, contact address, and mailing list address is alan@lxorguk.ukuu.org.uk

<ftp://linux.org.uk/pub/linux/alan/OLD-shadow.cabi.net/AberMUD5/SOURCE/>

LPMUD

The most popular combat-oriented MUD. Players cannot build. Be warned, though: LPMUD

servers version 3.* themselves are very generic - all of the universe rules and so forth are written in a separate module, called the mudlib. Most LPMUDs running are written to be some sort of combat system, which is why I've classified them here, but they don't have to be! Wizards can build onto the database, by means of an object-oriented C-like internal language called LP-C. It's named after its primary author, Lars Pensjö. Latest version is 3.2.1, aka Amylaar. Fairly stable, and size varies from medium to large. Driver (server) versions seem to have split into several main variants, not counting possible mudlibs (databases) available. Amylaar, CD, and MudOS are the current favorites. For further information, email to amyhaar@cs.tu-berlin.de.

There is a port of 3.1.2 for Amigas, called amud, now included in LPMUD v3.2. For further information email to mateese@ibr.cs.tu-bs.de.

See the [rec.games.mud.lp](#) FAQ for more info.

<ftp://lysator.liu.se:/pub/lpmud>

<ftp://nightfall.org>

<http://genesis.cs.chalmers.se>

CD Downloads: <http://genesis.cs.chalmers.se/downloads/>

<ftp://tu-bs.de:/pub/games/lpmud>

<ftp://ccs.neu.edu:/pub/mud/drivers/mudos>

There is a port of 3.1.2 for MSDOS, that requires at least a '386 to run. It accepts connections from serial ports.

<ftp://ccs.neu.edu:/pub/mud/drivers/lpmud/msdos>

DGD

Written by Felix Croes. A reimplementation from scratch of the LPMUD server. It is disk-based, and thus uses less memory. It's also smaller and lacks many of the features of the other LPMUD servers, though it is capable of simulating most of those features in LP-C. Many DGDs are simulating an LP, but there are several MUDs that now use DGD to simulate a MOO variant. The name stands for Dworkin's Generic Driver. Very stable. Runs on most variants of Unix, and has been ported to the Atari ST, Commodore Amiga, Macintosh, Windows NT, Windows 95, OS/2 and BeOS.

<ftp://imaginary.com:/pub/LPC/servers/DGD/>

<ftp://lysator.liu.se:/pub/lpmud/drivers/dgd>

DikuMUD

Newer than LPMud, and gaining in popularity. Almost identical from the players' point of view. Uses a guild system instead of a straight class system. Wizards can add on to the database, but there is no programming language, as in LP. It's named after the university at which it was written, Datalogisk Institut Københavns Universitet (Dept. of Datalogy, University of Copenhagen).

<coyote.cs.wmich.edu:/pub/Games/DikuMUD>

<ftp://envy.com:/pub/mud/servers>

<ftp://game.org:/pub/mud/diku>

Some Diku mud variants (Merc 2.2 and Envy 2.0) have been ported to Windows 95 and Windows NT.

NO KNOWN SITE

YAMA

PC mud writing system, using waterloo wattcp. Runs on a 640K PC/XT or better. Runs best with about a 1Mb ram disk, but is fine without. A separate windows version (yamaw) runs under windows and allows you to run a mud on a 286 or higher without taking over the machine.

sunacm.swan.ac.uk:/pub/misc/YAMA

UriMUD

Developed from an LPMud2.4.5, the code structure is very similar. Features include better speed, flexibility, stronger LPC, and the ability to handle multiple mudlibs under one parser. Latest version is 2.5.

urimud isp.net:/urimud/src

Ogham

From the players' point of view, similar to LPMUD. No programming language or database, as server and mudlib compile together to form a single binary executable. Latest version is 2.5.0.

ftp.ccs.neu.edu:/pub/mud/servers/ogham

CircleMUD

Derivative of DikuMUD Gamma v0.0. Developed by Jeremy Elson (jelson@cs.jhu.edu). Less buggy and tighter code all in all. Can be compiled under Win95/NT with Microsoft Visual C++, or with gcc on most Unix machines. Latest version is 3.0p12.

ftp.circlemud.org:/CircleMUD/
ftp2.circlemud.org:/CircleMUD/
ftp.cs.jhu.edu:/pub/CircleMUD

AmigaMUD

Written by scratch for Commodore Amiga computers. Includes custom client which supports graphics and sound. Disk based, fast programming language, standard scenario including built-in mail and bboards. Obtained from the Aminet ftp sites.

ftp.wustl.edu:/pub/aminet/game/role/AMCInt.lha, AMSrv.lha

Realms

Written by Andy Baillie for Amiga systems. Primarily combat based with races and classes. There are some social commands but not that many. The database may be modified both online and offline. It is disk based and uses caching to allow it to run on less powerful machines.

<http://www.babylon5.demon.co.uk/realms.html>

TinyMUD-style MUDs

TinyMUD

The first, and archetypical, socially-oriented MUD. It was inspired by and looks like the old VMS game Monster, by Rich Skrenta. Players can explore and build, with the basic @dig, @create, @open, @link, @unlink, @lock commands. Players cannot teleport, and couldn't use @chown or set things DARK until later versions. Recycling didn't exist till the later versions, either. It's called 'Tiny' because it is - compared to the combat-oriented MUDs. Original code written by Jim Aspnes. Last known version is 1.5.5. Not terribly big, and quite stable.

primerd.prime.com/pub/games/mud/tinymud

There is a PC port of TinyMUD, along with some extra code. It accepts connections from serial ports.

<ftp.tcp.com:/pub/mud/TinyMUD>

There is a modified version of TinyMUD called PRISM, that works for PCs, Atari STs, and most Unixes. It also comes with a internal BSX client for MSDOS.

<lister.cc.ic.ac.uk:/pub/prism>

TinyMUCK v1.*

The first derivative from TinyMUD. Identical to TinyMUD, except that it added the concept of moveable exits, called @actions. Also introduced the JUMP_OK flag, which allows players to use @teleport, and @recycle, which TinyMUD later added. Its name, MUCK, is derived from MUD, and means nothing in particular. Original code written by Stephen White. Latest stable verion is 1.2.c&r, which brought TinyMUCKv1 up to date with later TinyMUD things. Not terribly big. Please mail admin@mudconnect.com if you know the ftp location for this server.

TinyMUSH

The second derivative from TinyMUD. Also identical to TinyMUD, with the addition of a very primitive script-like language. Introduced JUMP_OK like TinyMUCK, and has recycling, except it is called @destroy. Also introduced the concept of PUPPETs, and other objects that can listen. In later versions the script language was extended greatly, adding math functions and many database functions. In the latest major version, 2.x, it's gone to a disk-basing system as well. Its name, MUSH, stands for Multi-User Shared Hallucination. Original code written by Larry Foard. The latest non- disk-based version is <PennMUSH> (see below) 1.7.2, which is quite similar to 2.* from the user's point of view. Both the disk-based version and the non-disk-based version are being developed at the same time. TinyMUSH is more efficient in some ways than TinyMUD, but winds up being larger because of programmed objects. Version 2.* in general uses less memory but a great deal more disk space. TinyMUSH 2.* and PennMUSH 1.7* both run under BSD and SysV. Most recent version of TinyMUSH is 2.2.4p4.

The yet-to-be-finished TinyMUSH 3.0 will be a combination of the latest versions of TinyMUSH and TinyMUX. See <http://www.godlike.com/tinymush-3.0/> for more information.

<ftp.tinymush.org:/pub/mud/tinymush>

<ftp.cis.upenn.edu:/pub/lwl>

<primerd.prime.com:/pub/games/mud/tinymush>

<ftp.tcp.com:/pub/mud/TinyMUSH>

There's also a port of 2.0.8p10 to Macintosh, currently at version 0.7.0d6.

TinyMUSH/Mac is written by Joshua Juran, and resides at <http://www.metamage.com/mush/>

PennMUSH

See [TinyMUSH](#) above. PennMUSH is a non-disk-based version of TinyMUSH, and is quite similar from the user's point of view. The latest version is 1.7.2, and will run under Unix, Win32 and Macintosh.

Website: <http://www.pennmush.org>

<ftp.pennmush.org/pub/PennMUSH/Source>

There is a port for Win32. Both executables and source are available for download.

<ftp.pennmush.org/pub/PennMUSH/Win32Binaries/>

There is a stable port for Macintosh at <http://mac.pennmush.org>

AlloyMUSH

AlloyMUSH is based on an early beta of TinyMUSH 2.2. It has added ANSI color, zones, powers, building functions, debug output redirection, and more. Latest version is 1.1p1.

<http://www.cris.com/~jmccgrew/mush>
<ftp.tinymush.org:/pub/mud/tinymush/src/alloy>

TinyMUCK v2.*

TinyMUCKv1.* with a programming language added. The language, MUF (multiple user forth), is only accessible to people with the MUCKER flag. Changed the rules of the JUMP_OK flag somewhat, to where it's nice and confusing now. MUF is very powerful, and can do just about anything a wizard can. Original version 2.* code written by Lachesis. Latest version is 2.3b, with several varieties (FBMUCK and DaemonMUCK 0.14 the most common). The name doesn't mean anything. Can be quite large, especially with many programs. Mostly stable.

<ftp.tcp.com:/pub/mud/TinyMUCK>

TinyMUSE

A derivative of TinyMUSH. Many more script-language extensions and flags. Reintroduced a class system, a-la combat-oriented MUDs. The name stands for Multi-User Simulation Environment. Latest official version is 1.8a4. Fairly stable.

<mcmuse.mc.maricopa.edu:/muse/server>

TinyMAGE

The bastard son of TinyMUSH and TinyMUCK. It combines some of MUSH's concepts (such as puppets, @adesc/@asucc, several programming functions, and a few flags) with TinyMUCK2.x. Interesting idea, really busted code. The name doesn't mean anything. Latest version is 1.1.2.

<ftp.tcp.com:/pub/mud/TinyMAGE>

MUG

Derivative of TinyMUD 1.4.1. Its name stands for Multi-User Game. Powerful but awkward

programming language, which is an extension of the user language; primitive notion of Puppets; inheritance; sane variable/property matching; arrays and dictionaries in hardcoded. Somewhat non-standard and buggy in a few places.

Requires gcc.2.4.5 or greater (or other good C++ compiler) to compile. Available by e-mail from wizard@cs.man.ac.uk; development site is UglyMUG (wyrm.compsoc.man.ac.uk 6239).

TeenyMUD

Originally a TinyMUD clone, written from scratch, with its main feature being that it was disk based. Original code written by Andrew Molitor. Now closer to a TinyMUSH, with some TinyMUCK influences. Latest version is 2.0.4betap3. Fairly small, and mostly stable.

<ftp://teleport.com:/pub/vendors/downsj>
<ftp://fido.econ.arizona.edu:/pub/teeny> <ftp://tinytymus.org:/pub/mud/teenymud>

TinyMUX

Originally a derivative of TinyMUSH 2.2 and mostly compatible with TinyMUSH 2.2, U1 and 3.0 as well as PennMUSH, it has continued to borrow and donate from the PennMUSH and TinyMUSH codebases. The latest version (2.0) is a thorough re-worked of the 1.6 version to be smaller, faster, and more stable. Win32 and Unix builds of the server are maintained simultaneously.

<http://svdltd.com/TinyMUX/>

Miscellaneous

MUD

The original, by Richard Bartle and Roy Trubshaw, was written back in 1978. An advanced version of MUD1 was, up until recently, running on CompuServe under the name of "British Legends". An internet-playable version will possibly be released soon.

MUD2 runs on Wireplay in the UK as well as on mud2.com. Although an internet version is not yet available it should be within a couple of months of this update (12/02/99).

UberMUD

The first MUD where the universe rules are written totally in the internal programming language, U. The language is very C/pascal-like. The permissions system is tricky, and writing up every universe rule (commands and all) without having big security holes is a pain. But it's one of the most flexible muds in existence. Great for writing up neat toys. It's also disk-based. Original code written by Marcus J Ranum. Latest version is 1.13. Small in memory, but can eat up disk space. Quite stable.

<ftp://decuac.dec.com:/pub/mud>
<ftp://white.toronto.edu:/pub/muds/uber>

MOO

An Object-Oriented MUD. Unfortunately, the first few versions weren't fully object oriented. Later versions fixed that problem. There is a C-like internal programming language, and it can be

a bit tricky. Original code written by Stephen White. Last version is 2.0a.

NO KNOWN SITE

LambdaMOO

An offshoot of MOO. Added more functionality, many new features, and a great deal more stability, in a general rewrite of the code. This is the only version of MOO that is still being developed, originally by Pavel Curtis, and now by Erik Ostrom. Latest version is 1.8.1.

<http://sourceforge.net/projects/lambdamoo/>

The MOO homepage is at <http://www.moo.mud.org/> and contains the MOO FAQ and LambdaMOO programmer's manual.

SMUG

Also known as TinyMUD v2.0. It has an internal programming language, and it does have some inheritance. Surprisingly similar to MOO in some ways. SMUG stands for Small Multi User Game. Original code written by Jim Aspnes.

<ftp.tcp.com:/pub/mud/Smug>

UnterMUD

A network-oriented MUD. It's disk-based, with a variety of db layers to choose from. An UnterMUD can connect directly to other UnterMUDs, and players can carry stuff with them when they tour the Unterverse. This can be a bit baffling to a new user, admittedly, but those people already familiar with the old cyberportals and how they work (invented way back with the original TinyMUD) will adjust to the new real cyberportals easily. There is both a primitive scripting language and much of the U language from UberMUD built in, as well as a combat system that can be compiled in if wanted. The parsing can be a bit odd, especially if you're used to the TinyMUD-style parser. Unter is also the only MUD that can run under BSD Unix, SysVr4 Unix, and VMS with MultiNet networking, with little to no hacking. Original code written by Marcus J Ranum.

Latest version is 2.1. Small in memory, but can eat up a lot of disk space.

<decuac.dec.com:/pub/mud>
<ftp.tcp.com:/pub/mud/UnterMUD>

Mordor

Most like a DikuMUD, with a built-in combat system, along with many choices for class and race, but not guild-based. Some "social-mud" features included as well. Also features online database editing as well as an offline db editor. Latest version is 4.61. Runs on BSD Unix, SysV Unix, NeXT Mach, IRIX, and WinNT & Win95. Written by Brett Vickers & Brooke Paul. Also comes with a custom client, [Muddle](#).

<mordor.nazgul.com:/pub/mordor>
<http://mordor.nazgul.com/>

COOLMUD

A distributed, object-oriented, programmable MUD server. Written by Stephen White.

<http://www.csclub.uwaterloo.ca/u/sfwhite/coolftp/>

Ursha Null 7

Ursha Null 7 is a Sci-Fi based graphical MUD/RPG. The server was designed by Russell T. Enderby and currently runs under DOS/Win9X/WinNT. The server supports both telnet and modem based connections. Support of RCP, ANSI, and ASCII connections.

It offers some unique features such as cellular vision phones that are a necessity for all players to have up to 4-way conferencing, voice mail, and a hand full of other options while in the game. Terminals are scattered throughout the game to interface to the Planetary Interactive Network (PIN).

But probably the most interesting feature is the sound effects and graphics throughout the game and during combat.

<http://www.ursha7.com/>

Cold Server

A server based on concepts behind MOO and CoolMUD. The server is disk-based and fast, and uses a proprietary programming language called ColdC.

Web site: <http://www.cold.org/>

FTP site: <ftp://ftp.cold.org/>

Note: just because we say something's available doesn't mean we have it. Please don't ask us; ask around for ftp sites that might have them, or try looking on ftp.tcp.com.

General Information

2.11. What do I do if my client/server won't compile?

Your first best bet is to check out the documentation and see if someone is listed as 'supporting' (i.e. generally responsible for) the program. If they are, send them a short, well-written e-mail note explaining your hardware and software completely as well as a transcript of the error. Do not post to the internet unless all other realistic options have been considered and taken -- generally speaking, most readers will not be interested in your dilemma and may get upset that you're wasting their time. Since MUDs have probably been compiled on every single platform since the Cyber 3000, there's a good chance that asking around the subculture will get you the answers you crave. Do not mail me. I probably won't know.

2.12. Should I read the documentation of whatever client or server I select?

Yes.

2.13. What is FTP, and how do I use it?

FTP stands for File Transfer Protocol, and is a way of copying files between networked computers. The best way to learn about ftp is to get the FTP FAQ, by emailing mail-server@rtfm.mit.edu with

send usenet/news.answers/ftp-list/faq

in the body of the message.

Not all ftps are alike, but here's a sample session on a unix system:

```
% ftp muds.okstate.edu
Connected to muds.okstate.edu.
220 muds.okstate.edu FTP server ready.
Name (muds.okstate.edu:jds): ftp <-- use 'ftp' as your login
331 Guest login ok, send ident as password.
Password: <-- use your email addr as pwd
230 Guest login ok, access restrictions apply.
ftp> cd pub/jds/clients <-- how to change directories
250 CWD command successful.
ftp> dir <-- ls also works
200 PORT command successful.
150 ASCII data connection for /bin/ls (139.78.112.6,4011) (0 bytes).
total 2310
-rw-r--r-- 1 4002 4002 34340 Feb 6 1992 amigaclient.lzh
...etc etc...
-rw-r--r-- 1 4002 4002 43093 Dec 13 1991 tinytalk.117.shar.Z
226 ASCII Transfer complete.
2631 bytes received in 0.7 seconds (3.6 Kbytes/s)
ftp> bin <-- VERY IMPORTANT! binary transfers
200 Type set to I.
ftp> get tinytalk.117.shar.Z <-- get filename
200 PORT command successful.
150 ASCII data connection for tinytalk.117.shar.Z (139.78.112.6,4012) (43093 bytes)
226 ASCII Transfer complete.
local: tinytalk.117.shar.Z remote: tinytalk.117.shar.Z
43336 bytes received in 0.28 seconds (1.5e+02 Kbytes/s)
ftp> bye <-- how to quit ftp
221 Goodbye.
%
```

Now that you've successfully ftped a file, you must unarchive it. There are many ways of archiving files; so many that they couldn't possibly all be listed here. In general, though, if a file ends in:

.Z	uncompress filename
.z	gunzip filename
.gz	gunzip filename
.tar	tar -xvf filename
.shar	sh filename
.zip	unzip filename

Generally, once you've unarchived your client or server, you must still compile it. This varies widely depending on the system you're on and the particular client or server. Your best bet is to look for a README or INSTALLATION file or something equally obvious, and then if you're still unsure, ask

someone locally to help you out.

If you are connecting directly to the Internet from your PC running Windows, or a Macintosh, you have it much simpler. Just use a FTP client (WS_FTP or CuteFTP for Windows) to connect to whichever server and download whichever client you want. For PC systems, look in this FAQ for clients which say they use Winsock.

This posting has been generated as a public service, but is still copyrighted 1996-1999 by Jennifer Smith. Modifications made after August, 1999 are copyrighted 1999 by Andrew Cowan. If you have any suggestions, questions, additions, comments or criticisms concerning this posting, contact Andrew Cowan (admin@mudconnect.com). Other Frequently Asked Questions (FAQ) postings contain information dealing with clients, servers, RWHO, and FTP sites. While these items aren't necessary, they are quite useful. I'd also like to thank cthonics (felixg@coop.com) for his help in writing these FAQs, ashne and Satoria for their help, and everyone else for helpful comments and suggestions. Thanks again to Alec Muffett (aem@aberystwyth.ac.uk) of alt.security.

The most recent versions of these FAQs are archived at <http://www.mudconnect.com/mudfaq/> and on rtfm.mit.edu in the news.answers archives.

Andrew Cowan / admin@mudconnect.com

FREQUENTLY ASKED QUESTIONS: Basic Information on RWHO and mudwho

This is part 3 in a 4 part series of FAQs.

Disclaimer: This document may be seen to be biased towards TinyMUDs. This is because the original author of this document mainly plays those types of servers, not because she thinks they are inherently better or worse than other types of servers. However, this document is meant to be generalized and useful for all MUDdom, and so corrections and contributions are always welcome. The new maintainers will be gradually modifying the FAQ to be geared towards all of the various server types.

Note: This section of the MUD FAQ is not currently being maintained. The two links included in section 3.3 are now obsolete and I was unable to find alternative locations. If anyone knows where to find new locations please email the FAQ maintainer and I will update this document accordingly.

Table of Contents

- [3.1. What is RWHO?](#)
- [3.2. How Does It All Work?](#)
- [3.3. Where Can I Get This Stuff?](#)
- [3.4. Where Are Some RWHO Servers?](#)

RWHO and mudwho

3.1. What is RWHO?

RWHO stands for *Remote WHO*. It's a way of getting a WHO list from a MUD, without even having to connect to that MUD at all. Anyone can get this output from a RWHO server (an *mwhod*), by using straight telnet to connect to a certain port (6889), or by using the client program *mudwho*. RWHO servers talk to other mwhods, passing information around, and are talked to directly by some MUDs, receiving information from them.

Any one mwhod keeps track of several MUDs, plus storing information passed it from other mwhods. Only MUDs that have the RWHO routines compiled in will be able to send their WHO list info to a mwhod. UnterMUDs have this capability built in; other MUDs have to have the routines installed first. The RWHO routines have been installed into TinyMUSH, TinyMUCK, LPMUD, DikuMUD, and AberMUD, as well as the Nightmare 2.5, TMI2, and Lima mudlibs for LPMUD without encountering any difficulty.

3.2. How Does It All Work?

mwhod is the RWHO server that runs on a particular host and keeps a list of known MUDs. It is initially primed with a list of "trusted" MUDs and passwords used for authentication, and will accept information about who is logged into those MUDs. The server also has a notion of a "peer" server, which can transfer it (occasionally) a copy of all of its list of who is logged on, and where. The idea is that the whole MUDding community could probably be served pretty well by about 5 peer mwhods that kept

each other up to date about what each one is seeing.

Communication between mwhods (and server updates sent to mwhods) is done with UDP datagrams, since they're fast, nonblocking, and throw-away. (RWHO information is considered to be interesting but not vital information, if you get my drift). Each MUD server only sends updates to a single mwhod, which may then propagate that information to its peers. This is done within the MUD server as follows:

- whenever the server boots, it sends a "hi there" packet to the mwhod, telling it that it's up and running.
- whenever a player connects, it sends a "so and so is here" packet to the mwhod, telling it that the user has connected.
- whenever a player disconnects, it sends a "so and so left" packet to the mwhod, telling it to delete the entry.
- every so often ("so often" being defined as a time agreed upon by the mwhod's owner, and the MUD's wizard, usually every 5 minutes or so) the MUD sends a "hi there" packet and a complete list of everyone that is on, just to refresh the mwhod's idea of who is logged into that MUD.

If a user connects to a specific port (6889) of a host machine running an mwhod they are given a formatted dump of the mwhod's current table of MUDs and players, and then disconnected. *mudwho* is a simple little program that contacts an mwhod and downloads this information. Ideally, the functionality of *mudwho* would be built into a player's client software, for ease of use. Two handy options can be used by *mudwho*, if the netlag to the mwhod server isn't too bad. The options are *-u <username>*, and *-m <mudname>*. If received before the timeout, the mwhod will then only dump WHO list information for the specified player or MUD.

The mwhod does some clever stuff as far as eventually timing information about of its tables - for example, if it hears absolutely nothing from a MUD for a certain amount of time, it will mark the MUD as down. Player entries are expired similarly. The design is based on the idea that we'll use UDP to just fling information out and hope it sticks, and then let the recipient clean it up, rather than to develop a more complex protocol based on TCPs and timeouts. To prevent a packet circular send situation, each entry that is sent is given a "generation" number, which is incremented by one each time it is forwarded along. In this manner, a MUD server might send a "so and so is here" (generation zero) to its local mwhod. The local mwhod will eventually send a copy to any peers it may have (generation one), and so forth. Part of the initial table that an mwhod uses to establish what peers it trusts contains a generation value, and it will neither accept nor propagate information to a specific peer that is of a higher generation value. This way, a "tree" of servers could theoretically be constructed, with the highest level one having a total view of a large MudIverse.

3.3. Where Can I Get This Stuff?

The RWHO routines can be ftp'd from decuac.dec.com, in pub/mud. Included is a HOW_TO file, which describes how to plug the routines into a MUD server, and also where to ask for a mwhod to use.

The mwhod program itself can also be found on decuac, but there is currently little need for another one running in the USA, except perhaps as a backup. There is, however, only one running in all of Europe, and further expansion may need to be made in that area.

3.4. Where Are Some RWHO Servers?

Updated 12/28/99 - we have been informed that the only known RWHO server is no longer running. So,

the only answer we can provide is that we do not know. If you have any information about RWHO servers that are still running please email us at admin@mudconnect.com.

This posting has been generated as a public service, but is still copyrighted 1996-1999 by Jennifer Smith. Modifications made after August, 1999 are copyrighted 1999 by Andrew Cowan. If you have any suggestions, questions, additions, comments or criticisms concerning this posting, contact Andrew Cowan (admin@mudconnect.com). Other Frequently Asked Questions (FAQ) postings contain information dealing with clients, servers, RWHO, and FTP sites. While these items aren't necessary, they are quite useful. I'd also like to thank cthonics (felixg@coop.com) for his help in writing these FAQs, ashne and Satoria for their help, and everyone else for helpful comments and suggestions. Thanks again to Alec Muffett (aem@aberystwyth.ac.uk) of alt.security.

The most recent versions of these FAQs are archived at
<http://www.mudconnect.com/mudfaq/> and on rtfm.mit.edu in the news.answers archives.

Andrew Cowan / admin@mudconnect.com